

LISTING OF CLAIMS

1. CANCELLED

2. (currently amended) **[[A]]** The method as in claim **[[1,]]** 3 further including populating a destination image with extracted contents of the source disk in which the destination image has ~~the identical~~ files, attributes, and structural relationships between files identical to files, attributes, and structural relationships between files of **[[as]]** the source disk.

3. (currently amended) A method as in claim 1, ~~further comprising, in for~~ creating an image of a source disk of a first computer on a ~~server~~ second computer that includes an operating system that has file system software that automatically detects the file system of disks mounted in the server computer, said method comprising:

~~creating a simulated source disk corresponding to the source disk;~~

mounting **[[the]]** a simulated source disk in the ~~server~~ second computer~~[[,]]~~ so that the file system software thereby automatically detecting the file system of the simulated source disk and therefore of the source disk and exposing the file system to software running on the server computer is accessible by the operating system as a local disk; and

configuring the simulated source disk as a proxy for the source disk by intercepting sector-based I/O requests directed to the simulated source disk and retrieving **[[the]]** source disk data from the source disk according to the intercepted sector-based I/O requests.

4. (currently amended) **[[A]]** The method as in claim 3, further comprising forwarding the intercepted sector-based I/O requests to the ~~source~~ first computer over a network.

1 5. (currently amended) **[[A]]** The method as in claim 4, in which the source disk
2 is associated with a ~~source~~ first computer that has a memory, further comprising:
3 loading an imaging client program in the memory of the ~~source~~ first computer, the
4 imaging client program not being resident on the source disk; and
5 passing the sector-based I/O requests to the imaging client program, the imaging
6 client program directing the intercepted sector-based I/O requests to the source disk.

1 6. (currently amended) **[[A]]** The method as in claim 5, further comprising:
2 loading a secondary operating system in the memory of the ~~source~~ first
3 computer, said secondary operating system not being present on the source disk and
4 mediating I/O requests between the imaging client program and the source disk.

1 7. (currently amended) **[[A]]** The method as in claim **[[3,]]** 2 further comprising:
2 mounting the destination image in an uninitialized state in the ~~server~~ second
3 computer as a simulated destination disk;
4 intercepting sector-based I/O requests directed to the simulated destination disk
5 and directing the contents of the intercepted sector-based I/O requests to the
6 destination image;
7 retrieving partition and file system layout information from the source disk;
8 formatting the simulated destination image to have the same partitioning and file
9 ~~system(s)~~ system as the simulated source disk and thus of the source disk; and
10 copying **[[the]]** files of at least one file system of the simulated source disk to the
11 corresponding file system of the simulated destination disk.

1 8. (currently amended) **[[A]]** The method as in claim 7, further comprising
2 converting the intercepted sector-based I/O requests to the simulated destination disk
3 into sector accesses within the destination image.

1 9. (currently amended) **[[A]]** The method as in claim 7, in which the destination
2 image is a virtual disk file associated with a virtual computer.

1 10. (currently amended) **[[A]]** The method as in claim 9, in which the ~~source~~
2 first computer is a physical computer and the source disk is a physical disk associated
3 with the physical computer.

1 11. (currently amended) **[[A]]** The method as in claim 9, in which the virtual
2 disk file is a sparse virtual disk, having a predetermined capacity and initial sector
3 contents with null values.

1 12. (currently amended) **[[A]]** The method as in claim 7, in which the source
2 disk is a source virtual disk.

1 13. (currently amended) **[[A]]** The method as in claim 12, in which the
2 destination disk is a physical disk.

1 14. (currently amended) **[[A]]** The method as in claim 7, in which the source
2 disk is a first virtual disk associated with a first virtual computer and the destination disk
3 is a second virtual disk associated with a second virtual computer.

1 15. (currently amended) **[[A]]** The method as in claim 7, in which the ~~source~~
2 first computer is the same as the ~~server~~ second computer.

1 16. (currently amended) A method for creating an image of a source disk of a
2 ~~source~~ first computer, in which contents of the source disk are arranged according to at
3 least one source file system, comprising:

4 in a ~~server~~ second computer that includes an operating system that has file
5 system software that ~~automatically~~ detects **[[the]]** a file system of disks mounted in the
6 ~~server~~ second computer, while the source disk is in an unmodified, unprepared state,
7 extracting the contents of the source disk, defining extracted contents, and populating a
8 destination image with the extracted contents of the source disk such that the
9 destination image may have a different sector-by-sector content than the source disk
10 but a destination file system logically equivalent to the at least one source file system,

with identical files, attributes, and structural relationships between files as the source disk;

~~creating a simulated source disk corresponding to the source disk;~~

mounting ~~[[the]]~~ a simulated source disk in the ~~server~~ second computer so that the simulated source disk is accessible by the operating system as a local disk, ~~the file system software thereby automatically detecting the file system of the simulated source disk and therefore of the source disk and exposing the file system to software running on the server computer;~~

configuring the simulated source disk as a proxy for the source disk by intercepting sector-based I/O requests directed to the simulated source disk and retrieving ~~[[the]]~~ source disk data from the source disk according to the intercepted sector-based I/O requests;

forwarding the intercepted sector-based I/O requests to the ~~source~~ first computer;

loading an imaging client program into a memory of the ~~source~~ first computer;

passing the intercepted sector-based I/O requests to the imaging client, the imaging client directing the intercepted sector-based I/O requests to the source disk;

~~operating system mediating, by the operating system, sector-based I/O requests~~ between the imaging client and the source disk;

mounting the destination image in an uninitialized state in the ~~server~~ second computer as a simulated destination disk;

intercepting sector-based I/O requests directed to the simulated destination disk and directing ~~[[the]]~~ results of the intercepted sector-based I/O requests to the destination image;

converting the intercepted sector-based I/O requests to the simulated destination disk into sector accesses within the destination image;

retrieving partition and file system layout information from the source disk;

formatting the simulated destination image to have the same partitioning and file system(s) as the simulated source disk and thus of the source disk; and

copying ~~[[the]]~~ files of at least one file system of the simulated source disk to the corresponding file system of the simulated destination disk.

1 17. CANCELLED

1 18. (currently amended) A system as in claim 17, ~~further for creating an image~~
2 ~~of a source disk in which contents of the source disk are arranged according to at least~~
3 ~~one source file system, said system comprising:~~
4 ~~a first computer having the source disk;~~
5 ~~a second computer having a memory with an operating system and an imaging~~
6 ~~server residing therein, the imaging server including computer executable instructions~~
7 ~~having code to create a simulated source disk that is a representation of information~~
8 ~~stored on the source disk and is accessed by the operating system as a local disk; and~~
9 ~~code to mount the simulated source disk in the second computer, with said memory~~
10 ~~including file system drivers to detect a file system of the simulated source disk and a~~
11 ~~network loopback driver intercepting sector-based I/O requests directed to the simulated~~
12 ~~source disk and retrieving source disk data from the source disk according to~~
13 ~~intercepted sector-based I/O requests intercepted by the network loopback driver,~~
14 ~~defining intercepted sector based I/O requests a server operating system that resides in~~
15 ~~the server computer;~~
16 ~~file system drivers within server operating system automatically detecting the file~~
17 ~~system(s) of disks mounted in the server computer;~~
18 ~~an imaging server running within the server computer and comprising computer-~~
19 ~~executable instructions:~~
20 ~~for creating a simulated source disk corresponding to the source disk;~~
21 ~~for mounting the simulated source disk in the server computer, the file system~~
22 ~~drivers thereby automatically detecting the file system of the simulated source disk and~~
23 ~~therefore of the source disk and exposing the file system to software running on the~~
24 ~~server computer; and~~
25 ~~a network loopback driver intercepting sector based I/O requests directed to the~~
26 ~~simulated source disk and retrieving the source disk data from the source disk~~
27 ~~according to the intercepted sector based I/O requests.~~

1 19. (currently amended) **[[A]]** The system as in claim 18, further comprising a
2 network adapter, residing in said memory, to forwarding forward the intercepted sector-
3 based I/O requests to the source first computer.

1 20. (currently amended) **[[A]]** The system as in claim 19, further comprising:
2 a first computer memory within the source first computer;
3 an imaging client installed in the first computer memory of the source computer;
4 said imaging client comprising computer-executable instructions that include code to
5 receive any source disk I/O requests issued from the second computer to the first
6 computer, code to direct the intercepted sector-based I/O requests to the source disk,
7 and code to pass the retrieved source disk data to the second computer in response to
8 the source disk I/O requests

9 ~~for receiving any source disk I/O requests issued from the server~~
10 ~~computer to the source computer,~~

11 ~~for directing the sector-based I/O requests to the source disk, and~~

12 ~~for passing the retrieved source disk data to the server computer in~~
13 ~~response to the source disk I/O requests.~~

1 21. (currently amended) **[[A]]** The system as in claim 18 wherein the imaging
2 server[[,]] further comprising: includes code to generate a simulated destination disk in
3 response to the second computer mounting the destination image, with said memory
4 further including a local loopback driver, a local adapter and a formatting module, with
5 the local loopback driver intercepting sector-based I/O requests directed to the
6 simulated destination disk and retrieving partition and file system layout information
7 from the source disk, the local adapter comprising code to convert the intercepted
8 sector-based I/O requests to the simulated destination disk into sector accesses within
9 the destination image and the formatting module comprising code to format the
10 destination image to have the same partitioning and file system(s) as the simulated
11 source disk and thus of the source disk, the imaging server having code to copy the files
12 of at least one file system of the simulated source disk to the corresponding file system
13 of the simulated destination disk

14 ~~a simulated destination disk generated by mounting the destination image in an~~
15 ~~uninitialized state in the server computer;~~
16 ~~a local loopback driver intercepting sector based I/O requests directed to the~~
17 ~~simulated destination disk and retrieving partition and file system layout information~~
18 ~~from the source disk;~~
19 ~~a local adapter comprising computer executable instructions for converting the~~
20 ~~sector based I/O requests to the simulated destination disk into sector accesses within~~
21 ~~the destination image; and~~
22 ~~a formatting module comprising computer executable instructions for formatting~~
23 ~~the destination image to have the same partitioning and file system(s) as the simulated~~
24 ~~source disk and thus of the source disk;~~
25 ~~the imaging server further comprising computer executable instructions for~~
26 ~~copying the files of at least one file system of the simulated source disk to the~~
27 ~~corresponding file system of the simulated destination disk.~~

1 22. (currently amended) **[[A]]** The system as in claim 21, in which the source
2 disk is a virtual disk.

1 23. (currently amended) **[[A]]** The system as in claim 22, in which the
2 destination disk is a physical disk.

1 24. (currently amended) **[[A]]** The system as in claim 21, in which the
2 destination image is a virtual disk file associated with a virtual computer.

1 25. (currently amended) **[[A]]** The system as in claim 24, in which the ~~source~~
2 first computer is a physical computer and the source disk is a physical disk associated
3 with the physical computer.

1 26. CANCELLED.

1 27. (currently amended) A system for creating an image of a source disk of a
2 ~~source~~ first computer, which has a memory and in which contents of the source disk are
3 arranged according to at least one source file system, comprising:

4 a ~~server~~ second computer;

5 a server operating system that resides in the ~~server~~ second computer;

6 file system drivers within the server operating system automatically detecting
7 ~~[[the]]~~ at least one file system(s) system of disks mounted in the ~~server~~ second
8 computer;

9 an imaging server running within the ~~server~~ second computer and comprising
10 computer-executable instructions:

11 for extracting the contents of the source disk, defining extracted contents,
12 and populating a destination image with the extracted contents of the source disk such
13 that the destination image may have a different sector-by-sector content than the source
14 disk but a destination file system logically equivalent to the at least one source file
15 system;

16 for creating a simulated source disk corresponding to the source disk;

17 while the source disk is in an unmodified, unprepared state, for mounting
18 the simulated source disk in the ~~server~~ second computer, the file system drivers thereby
19 automatically detecting the file system of the simulated source disk and therefore of the
20 source disk and exposing the file system to software running on the ~~server~~ second
21 computer;

22 a network loopback driver intercepting sector-based I/O requests directed to the
23 simulated source disk;

24 a network adapter forwarding the intercepted sector-based I/O requests to the
25 ~~source~~ first computer;

26 an imaging client installed in the memory of the ~~source~~ first computer, said
27 imaging client comprising computer-executable instructions

28 for receiving any source disk I/O requests issued from the ~~server~~ second
29 computer to the ~~source~~ first computer,

30 for directing the intercepted sector-based I/O requests to the source disk,
31 and

32 for passing ~~the retrieved source disk data~~ to the ~~server~~ second computer
33 source disk data retrieved in response to the source disk I/O requests;
34 a simulated destination disk generated by mounting the destination image in an
35 uninitialized state in the ~~server~~ second computer;
36 a local loopback driver intercepting sector-based I/O requests directed to the
37 simulated destination disk and retrieving partition and file system layout information
38 from the source disk;
39 a local adapter comprising computer-executable instructions for converting the
40 intercepted sector-based I/O requests to the simulated destination disk into sector
41 accesses within the destination image; **[[and]]**
42 a formatting module comprising computer-executable instructions for formatting
43 the destination image to have the same partitioning and file ~~system(s)~~ system as the
44 simulated source disk and thus of the source disk; and
45 the imaging server further comprising computer-executable instructions for
46 copying **[[the]]** files of at least one file system of the simulated source disk to the
47 corresponding file system of the simulated destination disk.

1 28. (New) A tangible medium embodying instructions causing a universal
2 computer management system (UCMS) to perform a method, the UCMS being a
3 general purpose computer system, the method comprising:
4 registering a plurality of computer systems that are in electronic communication
5 with the UCMS, each of the registered computer systems being one of a physical
6 computer system or a virtual computer system, wherein each virtual computer system
7 comprises a software abstraction of a physical computer system, wherein a software
8 layer underlying the virtual machine exports an interface to the virtual machine that is
9 equivalent to an interface presented directly by hardware of a computer system;
10 maintaining a registration database to store information related to each of the
11 registered computer systems, the information comprising hardware configuration of
12 each of the registered computer systems and a deployment state of each of the
13 registered computer systems;
14 deploying an image accessible to the UCMS to a selected one of the registered
15 computers, the image being a disk image containing a software stack including an
16 installed operating system that is configured to function with the hardware configuration
17 of the selected registered computer, the deploying comprising copying the image to a
18 physical disk attached to the selected registered computer; and
19 updating the registration database to reflect the deployment of the image to the
20 selected registered computer.

1 29. (New) The tangible medium of claim 28, wherein the method further
2 comprises registering configurations that are common to a plurality of the registered
3 computer systems as separate entries in the registration database hardware.

1 30. (New) The tangible medium of claim 28, wherein the UCMS includes an
2 imaging server for performing the deploying of the image and the selected registered
3 computer comprises a remote physical computer system.

1 31. (New) The tangible medium of claim 28, wherein the selected registered
2 computer comprises a virtual computer system executing on the UCMS and the

3 deploying of the image comprises making a local copy of the image, wherein the local
4 copy becoming an active image of the selected registered computer.

1 32. (New) The tangible medium of claim 28, wherein the selected registered
2 computer comprises a virtual computer system executing on a remote physical host
3 system, the physical disk being attached to the selected registered computer comprising
4 a disk attached to the remote physical host system.

1 33. (New) The tangible medium of claim 28, wherein the method further
2 comprises:
3 detecting a presence of a computer on a network in communication with UCMS
4 by communicating with an agent executing in a secondary software stack of the
5 computer;
6 communicating with an imaging client running in a secondary software stack of
7 the detected computer; and
8 retrieving hardware configuration of the detected computer from the imaging
9 client.

1 34. (New) The tangible medium of claim 33, wherein the method further
2 comprises:
3 determining that the detected computer is not one of the plurality of registered
4 computers;
5 adding the detected computer to a list of discovered but unregistered computer
6 systems; and
7 providing an alert for an administrator to notify the administrator of the presence
8 of the detected computer.

1 35. (New) The tangible medium of claim 28, wherein the deploying of the
2 image comprises consulting a resource database to identify a source location of the
3 image file, the source location being a location accessible by the UCMS.

1 36. (New) The tangible medium of claim 28, wherein the deploying of the
2 image further comprises:
3 obtaining a hardware configuration of the image, the hardware configuration of
4 the image comprising attributes of hardware with which the software stack of the image
5 is compatible;
6 comparing the hardware configuration of the selected registered computer with
7 the hardware configuration of the image; and
8 performing the deploying of the image without modifying the image when the
9 hardware configuration of the selected registered computer matches the hardware
10 configuration of the image.

1 37. (New) The tangible medium of claim 36, wherein the obtaining of the
2 hardware configuration of the image comprises accessing an image record from a
3 resources database, the image record being associated with the image.

1 38. (New) The tangible medium of claim 36, wherein the obtaining of the
2 hardware configuration of the image comprises inspecting the image being deployed.

1 39. (New) The tangible medium of claim 38, wherein the inspecting of the
2 image comprises performing a temporary loop-back mount of the image, thereby
3 allowing a file system of the image to be accessed by the UCMS through a mapped
4 simulated local disk.

1 40. (New) The tangible medium of claim 39, wherein the inspecting of the
2 image further comprises accessing a registry drive maintained by the software stack
3 contained by the image.

1 41. (New) The tangible medium of claim 40, wherein the accessing of the
2 registry drive comprises:
3 invoking an API to load the registry drive from the image into a temporary subtree
4 within a registry drive of the operating system of the UCMS; and

5 accessing the ones of the registry entries indicative of the hardware configuration
6 of the image using an API function provided by the operating system of the UCMS.

1 42. (New) The tangible medium of claim 38, wherein the inspecting of the
2 image comprises inspecting the external characteristics of a file, the external
3 characteristics including name and file type.

1 43. (New) The tangible medium of claim 38, wherein the inspecting of the
2 image comprises reading internal contents of a file.

1 44. (New) The tangible medium of claim 43, wherein the inspecting of the
2 image further comprises computing a signature of the file and comparing the signature
3 to signatures of known file variants, wherein an identification of a matching signature is
4 indicative of at least one of the attributes of the hardware configuration of the image.

1 45. (New) The tangible medium of claim 38, wherein the deploying further
2 comprises:
3 reconfiguring contents of the image when the hardware configuration of the
4 selected registered computer does not match the hardware configuration of the image,
5 the reconfiguring comprising performing a loop-back mount of the image to allow the
6 UCSM to access and manipulate contents of the image through a mapped simulated
7 disk, the reconfiguring further comprising writing necessary changes to the image so
8 that the software stack is compatible with the hardware configuration of the selected
9 registered computer.

1 46. (New) The tangible medium of claim 45, wherein the changes to the image
2 are written to a temporary redo log.

1 47. (New) The tangible medium of claim 45, wherein the reconfiguring of the
2 contents of the image comprises:

3 replacing appropriate operating system files with substitute files extracted from
4 containers in a file cache accessible to the UCMS.

1 48. (New) The tangible medium of claim 47, wherein the substitute files are
2 identified by:

3 determining a current patch level for the image, the patch level relating to a
4 version of system files installed on the image; and

5 for each of the operating system files being replaced corresponding to a current
6 patch level, identifying a variant operating system file that is up-to-date for the current
7 patch level, the current patch level being a current file state of an operating system as a
8 function of a most recently applied set of patches.

1 49. (New) The tangible medium of claim 45, wherein a difference between the
2 hardware configuration of the selected registered computer and the hardware
3 configuration of the image comprises a difference in devices that are not critical to a
4 boot process for the software stack.

1 50. (New) The tangible medium of claim 49 wherein the difference in devices
2 comprises a difference in an image configuration for a hardware address of an attached
3 network interface card (NIC) and an actual hardware address of an attached NIC of the
4 selected registered computer and the necessary changes to the image comprise
5 creating or modifying binding settings for the NIC.

1 51. (New) The tangible medium of claim 28, wherein the method further
2 comprises:

3 after registering a new one of the hardware configurations, determining whether
4 a set of substitute files needed to reconfigure an image to support the new hardware
5 configuration are present in a locally accessible cache;

6 obtaining missing ones of the substitute files via a network connection when the
7 set of substitute files is not present.

52. (New) The tangible medium of claim 51, wherein the obtaining of the missing ones of the substitute files comprises prompting a user to download software containers containing the missing files via the Internet and into the locally accessible cache.